REMARKS

Applicants gratefully acknowledge Examiner Ngo for taking time on June 5, 2007, to participate in a telephone interview including co-inventor Gustavson, who presented a summary of the present invention and its distinction from the cited reference Myzewski (US Patent No. 5,099,447). The statutory subject matter rejection was also discussed. The Examiner indicated that he would consider points brought up by the presentation in this next evaluation and

Claims 1, 2, and 4-27 are all the claims presently pending in the application. Claim 3 is canceled above.

It is noted that composite blocking is the more generic concept of double blocking addressed in claims 4, 11, 17, 20, 24, and 27, which claims are not rejected under the prior art evaluation currently of record. Applicants, therefore, conclude that the Examiner considers these claims to be allowable pending resolution of the issue of non-statutory subject matter.

It is noted that Applicants specifically state that no amendment to any claim herein should be construed as a disclaimer of any interest in or right to an equivalent of any element or feature of the amended claim.

Claims 1, 2, and 4-27 stand rejected under 35 U.S.C. § 101 as allegedly directed to non-statutory subject matter. Claims 1, 2, 5-10, 12-16, 18, 19, 21-23, and 25-27 stand rejected under 35 U.S.C. § 103(a) as unpatentable over U.S. Patent No. 5,099,447 to Myszewski, further in view of US Patent Publication 2003/0088600 to Lao et al.

These rejections are respectfully traversed in the following discussion.

I.   THE CLAIMED INVENTION

The claimed invention is directed to method of increasing at least one of efficiency and speed in executing a matrix subroutine on a computer. Data for a matrix subroutine call is stored as contiguous data in a computer memory in an increment block size that is based on a cache size of the computer. A first dimension of the block is larger than a corresponding first dimension of the cache and a second dimension of the block is smaller than a corresponding second dimension of the cache, so that the block fits into a working space of the cache.

Since standard Fortran and C two-dimensional arrays are stored in memory/processed in either a column or a row of the matrix data, depending on which language is used, there are inefficiencies in retrieving data for sub-blocks of matrix data, as discussed in more detail below.

In contrast, the present invention teaches that the data is to be stored in memory as a collection of sub-blocks of data that each fits into the cache working area and that is stored contiguously in memory, to preclude thrashing of data in cache.

By actually storing in memory the matrix data as contiguous submatrix data and by sizing the submatrix block of data so that one dimension of the submatrix block exceeds the dimension of the cache relative to the size of the cache (a number of submatrix blocks make up a square matrix), the processing is executed faster than conventional methods, since data thrashing is reduced or eliminated.

Another aspect of the present invention is the recognition that many engineering/ scientific matrices are typically directed toward square matrices, since such square matrices express linear equations in which a specific solution can be found. In contrast to conventional wisdom, the present invention also teaches that such square matrices can be viewed as rectangular blocks of matrix data that can be brought into a working area of cache as a block of contiguous data, again, thereby reducing or eliminating the thrashing that typically occurs when the entire square block is retrieved. The composite block of the present invention allows matrix data that, in modern engineering/scientific applications, typically involve a matrix having at least one dimension that exceeds either dimension of the working area of the cache, to be brought into cache in contiguous blocks that address the problem that plagues the art of cache data thrashing.

As an example of increased processing efficiency, an application related to the square blocking of the present invention using contiguous DGEMM operands is Cholesky factorization. Cholesky factorization is an example of a DLAFA. On the IBM Power3 processor the implementation of Cholesky factorization using the present invention achieves 92% of peak performance whereas conventional full format LAPACK DPOTRF achieves 77% of peak performance. Moreover, all programming for the new data structures discussed in the present invention can be accomplished in standard Fortran (or C/C++), through the use of higher dimensional full format arrays. Thus, no new compiler support is necessary to implement the present invention.

II.   THE 35 USC §101 REJECTIONS

Claims 12, and 4-27 stand rejected under 35 U.S.C. §101 as allegedly directed toward non-statutory subject matter.

In paragraph 7 on page 5 of the Office Action the Examiner explains his rationale for maintaining this rejection:

"Regarding the rejection under 35 USC 101, it is respectfully submitted that a method of executing a matrix subroutine, or computer or program implementing the method, if not limited to a practical application to reduce a useful, concrete, and tangible result would [be] directed to a non-statutory subject matter since it merely involve[s] calculations and manipulations of data values and to produce a result also data values. Since the executing [of] a matrix subroutine itself is non-statutory regardless how it is implemented, an improvement in executing the method is also non-statutory. Regarding claims 15-18, since the claimed invention is not limited to a computer readable storage medium, the recited "a signal bearing medium" clearly includes carrier signal and non computer readable storage medium, which are non-statutory subject matter."

Applicants respectfully traverse the above-recited Examiner's reasoning, as follows.

First, relative to claims 15-18, although Applicants disagree with the above Examiner's characterization, these claims have been reworded to clearly recite Beauregard language, as found in US Patent 5,710,578, issued on January 20, 1998, to Beauregard, et al., and the subject of the holding of *In re Beauregard*, 53 F.3d 1583 (Fed. Cir.1995). As Applicants explained in their previous response, in that case, the Commissioner of the USPTO requested that the Court dismiss Beauregard's appeal to the Federal Circuit because the USPTO had changed its mind on the issue of the claimed invention. That is, the Commissioner stated to the court "... *that computer programs embodied in a tangible medium, such as floppy diskettes, are patentable subject matter under 35 U.S.C. §101 and must be examined under 35 U.S.C. §§ 102 and 103."* Therefore, Applicants submit that claims 15-18 are clearly statutory subject matter by reason of the USPTO's actions in this case and the subsequent patent issued to Beauregard.

Second, relative to the Examiner's contention that "... *a method of executing a matrix subroutine, or computer or program implementing the method, if not limited to a practical application to reduce a useful, concrete, and tangible result would [be] directed to a non-*

statutory subject matter since it merely involve[s] calculations and manipulations of data values and to produce a result also data values", Applicants request that the Examiner provide a case holding that supports this position.

Third, and more significant, Applicants again point out that the present invention is directed to the storage of data in a size and format that increases the speed and efficiency of the processing. It is clearly related to size of the cache of the machine executing the processing. It is not directed to defining the mathematical process involved in the processing. As such, the present invention clearly is directed to the mathematical processing only to the extent that the processing was preceded by the data format described in the independent claims on a specific machine. There are no claims that define a mathematical algorithm.

Applicants submit that, contrary to the Examiner's characterizations on the record, the only reluctance of recent case holdings relative to mathematics is whether a mathematical algorithm is being preempted by a claimed invention. Such preemption is clearly not occurring in the claimed invention of the present application, since the underlying mathematical algorithm, presuming *arguendo* that the matrix processing is regarded as a "mathematical algorithm", can still be executed in the inefficient manner of the conventional method, including the accompanying thrashing. Thus, there clearly is no preemption of any mathematical algorithm by the claimed invention.

More important, Applicants submit that the increased speed and efficiency resultant from the present invention inherently provides a real-world result, thereby satisfying the "useful, concrete and tangible result" test, should such test be deemed appropriate for the present method claims.

As best understood, the Examiner's sole reason for continuing to reject the present invention as non-statutory is based upon the Examiner's characterization in the rejection that "…*the claims merely involve[s] calculations and manipulations of data…. The inputs are numbers and the results are also numbers. Further, the result of the invention is merely numerical values without a practical application recited in the claims. It is not [a] real world result, and thus is not useful, concrete and tangible. Therefore, the claimed invention is directed to non-statutory subject matter as the claims fail to accomplish a practical application.*"

In response, Applicants would possibly agree with the Examiner if the present invention were directed to defining the mathematical operation being executed by the processing of the

matrix data. In that case, the Examiner would arguably have a valid point that the invention involves only "calculations and manipulations of data" and might well be non-statutory.

However, as previously explained, the present invention is <u>not</u> directed to defining the mathematical processing. Rather, the present invention is directed to <u>preliminarily organizing the data</u> to be used in the matrix operation so that the <u>processing is faster and more efficient</u>. Thus, contrary to the Examiner's characterization, the present invention <u>does indeed</u> provide a <u>real world result</u> and is <u>not</u> "merely providing numbers having no practical application."

The Examiner's confusion stems from the failure to recognize that the present invention is not defining or claiming the underlying mathematical processing of matrix data. Rather, it is directed to <u>improving the speed and efficiency of the processing</u>, which is inherently a real world result that is useful, concrete and tangible.

Stated slightly differently, the Examiner improperly attempts to consider that anything related to a processing involving a mathematical operation is non-statutory unless the mathematical operation is defined as involved in a practical application. Applicants respectfully submit that such articulation is not case law and, again, requests that the Examiner <u>provide a citation supporting his position</u>, so that Applicants can distinguish the facts of that holding.

Indeed, Applicants submit that it is well established that an invention involving a mathematical processing is statutory as long as the invention is not directed to preempting a mathematical algorithm.

Even more important, the method of the present invention is <u>not</u> a mathematical algorithm and does <u>not</u> preempt any mathematical algorithm. Rather, the method of the present invention is the <u>storage of data in a specific block size of data that is related to the size of cache executing the mathematical processing</u>. The <u>result</u> of the method of the present invention is clearly an <u>increase in speed and efficiency of the processing on this machine</u>, which result is <u>inherently</u> statutory subject matter, since it inherently involves a real world result.

Finally, Applicants again submit that, <u>even if the method claims are deemed non-statutory</u> under the "useful, tangible and concrete result" test, the <u>apparatus</u> claims (e.g., claims 10-14) and <u>Beauregard</u> claims (e.g., claims 15-18) would still be statutory, as being directed to a machine and product of manufacture, respectively, both of which are categories expressly identified in 35 USC §101.

In view of the foregoing, the Examiner is respectfully requested to reconsider and

withdraw these rejections for non-statutory subject matter.

III. THE REJECTION UNDER 37 CFR §112, SECOND PARAGRAPH

Claims 1, 2, 5-10, 12-16, 18, 19, 21-23, and 25-27 stand rejected under 37 CFR §112, second paragraph, because the Examiner considers that the claim language "… storing data contiguously … in an increment block size …" is somehow unclear, since, the Examiner continues "… data of a matrix contiguously stored in memory can be view as being stored in any increment size."

Applicants respectfully traverse this characterization and this rejection. In contrast to matrix data as typically stored in memory, the present invention teaches to bring an entire block of data into cache, where the block size has been predetermined to fit into the cache working area even though the block has one dimension larger than the cache working area. There is no suggestion in the references of record to place matrix data in contiguous order in blocks of data to be moved into cache in units of these blocks having the dimensions described in the independent claims.

Applicants further submit that, if the cited wording of the claim is unclear in the manner alleged by the Examiner, the only reason for such lack of clarity is that the Examiner ignores the claim wording that provides the clarification to address the Examiner's concern.

That is, the remainder of the claim limitation contains the information that appropriately addresses the Examiner's concern by further clarifying the block size relative to the cache size.

Therefore, Applicants respectfully submit that there is no lack of clarity in the claim wording when viewed in its entirety and respectfully request that the Examiner reconsider and withdraw this rejection.

IV. THE PRIOR ART REJECTION

The Examiner alleges that Myszewski, when modified by Lao, renders obvious the claimed invention described by claims 1, 2, 5-10, 12-16, 18, 19, 21-23, and 25-27. Applicants respectfully disagree and submit that the rejection of record fails to meet the initial burden of a *prima facie* obviousness rejection, since, even if the two cited references were to be combined,

the combination would still fail to satisfy the plain meaning of the claim language of even the independent claims.

That is, the present invention describes a general technique of dividing data into large data blocks related to the size of the working area of the cache. Moreover, as clearly described in the independent claims, the blocks of data of the present invention involve blocks having <u>one dimension larger than the cache dimension and one dimension smaller than the cache dimen</u>sion. Neither reference cited by the Examiner suggests a block size with such dimensions and the Examiner makes no attempt to point to any description to support the rejection. The description in paragraph [0071] of secondary reference Lao and the example therein does <u>not</u> satisfy the description of the block size described in the independent claims, since the partitions shown are related to the matrix size, not the cache size. Nor is there is no suggestion in paragraph [0071] to have a block dimension larger than the cache dimension.

Hence, turning to the clear language of the claims in neither Myszewski nor Lao is there any teaching or suggestion of: "…a first dimension of said block being <u>larger than</u> a corresponding first dimension of said cache and a second dimension of said block being <u>smaller than</u> a corresponding second dimension of said cache, <u>such that said block fits into a working space of said cache</u>", as required by independent claim 1. The remaining independent claims have similar language.

Therefore, Applicants submit that there are elements of the claimed invention that are neither taught nor suggested in the prior art of record, and the Examiner is respectfully requested to reconsider and withdraw this rejection.

Additionally, Applicants submit again the following comments relative to this prior art rejection, particularly since the prior art rejection does not address all claims.

The present invention teaches the technique of composite blocking, as articulated in the independent claims. The "double blocking" defined in claims 4, 11, 17, and 20 is a specific embodiment of composite blocking. There is <u>no</u> suggestion in Myszewski to convert and <u>store in memory</u> the matrix data of a matrix in square blocks larger than cache and consisting of rectangular blocks of contiguous data that will each fit into the working area of a cache (even though one dimension of the rectangular blocks is larger that its corresponding cache dimension).

To consider this prior art reference in a different perspective, a <u>rectangular block makes DGEMM run faster</u>. However, <u>the DLAFA requires square blocking</u>. The composite blocking

of the present invention serves both purposes. Myszewski does not even recognize this disparate requirement and does not suggest a technique that satisfies this dual requirement, nor does it address DLAFA.

The technique of the present invention reduces the data thrashing that typically occurs when matrix data is simply retrieved in its original format as is required by higher level languages such as Fortran and C.

The data thrashing occurs, in the conventional processing, because portions of the matrix data are typically flushed from cache during processing so that additional memory accesses are required as the flushed data becomes needed for current processing and because matrix data is typically not contiguous as actually stored in memory. The present invention addresses this data thrashing by actually storing in memory the matrix data as rectangular contiguous sub blocks of a size that fits into cache and, if the matrix data is also contiguous, it will also be organized on memory line size, thereby increasing the speed at which future matrix data will enter the cache to be consumed in the matrix processing. This occurs because the data actually brought into cache from memory will be completely contiguous matrix data (rather than standard layout data in portions of some lines of data) and because the matrix subroutine processing will fit the processing result back into the data currently being consumed, rather than inadvertently flushing out matrix data not yet being processed.

In contrast to the conventional method of dealing with matrix data, the present invention teaches storing the matrix data in memory as actually being rectangular sub blocks of data that fits the sub blocks together so that all the matrix data is contiguous, where "contiguous" means that the data is in the order to be used in its processing. The standard format of higher level conventional computer languages makes no attempt to place matrix data into its appropriate "contiguous" format.

Moreover, Applicants again submit that the rejection currently of record also fails to point to specific lines and columns in Myszewski that satisfy the plain meaning of the language of any of the rejected dependent claims, and respectfully requests that the Examiner provide specific reference to line/columns prior to proceeding to Appeal. The rejection currently of record simply makes conclusory statements that are not supported in the reference, since the present invention is actually dealing with inefficiencies caused by storing data as required by higher level language implementation of matrix processing, as typically done in Fortran or C,

where data must be stored in standard format.

      The Myzewski patent describes standard cache blocking. The ideas behind standard cache blocking were first discovered and reduced to practice during 1984 and 1985 by the IBM Engineering and Scientific Subroutine Library (ESSL) programming team. In February 1986 IBM released the ESSL programming library to the public. The designers of ESSL recognized the need for level 3 BLAS and helped to propose as an Industry Standard the Level 3 BLAS. ESSL had a version of DGEMM even before the standard was adopted.

      Myzewski does not mention any format changes to DGEMM's operands, and analysis of his codes proves that he is using the formats required by DGEMM. It is a well known fact that certain DGEMM operands will thrash any L1 cache. By using the data structure of the present invention, one can prove that it minimizes L1, L2, and TLB misses.

      Therefore, in its exemplary embodiment, the present invention is directed at improving the performance of DLAFA. In the high performance DLA community, BLAS 3 were invented to make DLAFA run faster. To avoid the performance flaw described above, one can use the new data structure (e.g., memory storage technique) described in the present invention. Composite blocking, the subject of the present invention, is a means to increase DLAFA performance when using Square Blocks (SB). Note that a SB is a fundamental building block of the data structure. An innovation in the present invention is to realize that a SB can also be viewed as a collection of rectangular contiguous blocks, each fitting into the working area of L1 cache (typically, the major area of cache), even though the rectangular block has one dimension that is larger than the corresponding cache dimension. Also, rectangular blocks are preferred by DGEMM kernels, as its function C = C - AB has an asymmetry.

      Moreover, one exemplary key feature of the present invention is that input operands of all DGEMM calls be stored as contiguous data in memory. A major use of the present invention is for producing high performance Dense Linear Algebra Factorization Algorithms (DLAFA), since the standard full format data structures of Dense Linear Algebra (DLA) hurt the performance of its factorization algorithms. Full format rectangular matrices are the input and output of the level 3 BLAS.

      It follows, therefore, that the LAPACK and Level 3 BLAS approach has a basic performance flaw. By using composite blocking described in the present invention, this flaw can be removed.

In contrast, Myszewski was directed towards standard cache blocking, since the inputs to BLAS3 DGEMM are standard full format arrays.

As mentioned earlier, measurable improvement in efficiency has been measured by using the present invention for Cholesky factorization as an example of a DLAFA, wherein, on the IBM Power3 processor Cholesky factorization achieves 92% of peak performance, whereas conventional full format LAPACK DPOTRF achieves 77% of peak performance.

Therefore, Applicants submit that there are elements of the claimed invention that are not taught or suggested by Myszewski, even if modified by Lao, and the Examiner is respectfully requested to withdraw this rejection.

V. FORMAL MATTERS AND CONCLUSION

In view of the foregoing, Applicant submits that claims 1, 2, and 4-27, all the claims presently pending in the application, are patentably distinct over the prior art of record and are in condition for allowance. The Examiner is respectfully requested to pass the above application to issue at the earliest possible time.

Should the Examiner find the application to be other than in condition for allowance, the Examiner is requested to contact the undersigned at the local telephone number listed below to discuss any other changes deemed necessary in a telephonic or personal interview.

The Commissioner is hereby authorized to charge any deficiency in fees or to credit any overpayment in fees to Assignee's Deposit Account No. 50-0510.

Respectfully Submitted,
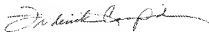
Date: August 21, 2007

Frederick E. Cooperrider
Registration No. 36,769

McGinn Intellectual Property Law Group, PLLC
8321 Old Courthouse Road, Suite 200
Vienna, VA 22182-3817
(703) 761-4100
Customer No. 21254

17

CERTIFICATION OF TRANSMISSION

    I certify that I transmitted electronically, via EFS, this Amendment under 37 CFR §1.111 to Examiner C. Ngo on August 21, 2007.

_____
Frederick E. Cooperrider
Registration No. 36,769